

Appl. No.: 09/611,920
Amdt. dated March 10, 2004
Reply to Office Action of December 18, 2003

Amendments to the Specification:

Please replace the paragraph beginning at page 3, line 15 through page 4, line 7, with the following rewritten paragraph:

A1
Turning now to the drawings, and specifically referring to Figure 1, a system 100 for providing LDAP directory server access to a plurality of client server applications is shown in accordance with the prior art. Typically, when a client server application 120 desires to access data from a directory server 110, the application 120 establishes a direct connection 125 to the directory server 110 through a binding operation. The function of the binding operation is to initiate a protocol session between a client and the directory server 110, and to allow authentication of the client to the server 110. The client server application 120 establishes the connection 125 to the server 110 via a bind request. Upon receiving the bind request from the client server application 120, the directory server 110 will authenticate the requesting client, if necessary, and attempt to set up a protocol session with the client. The directory server 110 subsequently sends a bind response to the client server application 120, thereby providing an indication of the status of the session startup request. Upon successfully establishing the connection 2125 with the directory server 110, the application 120 then retrieves the desired data from the directory server 110 by performing a search operation. After retrieving this desired data, the client server application 120 may then perform an unbind operation to terminate the protocol session between the client server application 120 and the directory server 110. Subsequent to receiving the unbind request, the directory server 110 closes the connection 2125 with the client server application 120.

Please replace the paragraph beginning on page 4, line 23 through page 4, line 14, with the following rewritten paragraph:

A2
As previously discussed, the client server applications 120 running on the OS typically establish a direct connection to the directory server 110, thereby placing a substantial load on the directory server 110 by engaging individual connections with each application 120. As a result of this additional load, the

Appl. No.: 09/611,920
Amdt. dated March 10, 2004
Reply to Office Action of December 18, 2003

A2
directory server 110 suffers substantial degradation in performance for data retrieval. To alleviate the burden on the directory server 110, an LDAP caching daemon 210 is provided for connection to the directory server 110 in accordance with one embodiment of the present invention. In the illustrated embodiment, the LDAP caching daemon 210 is a multi-threaded Internet UNIX daemon, and accesses data from the directory server 110 via a plurality of connections represented at 215. The number of connections at 215 may be configurable to facilitate the retrieval of data from the directory server 110 by the caching daemon 210. Furthermore, in addition to being coupled to the one directory server 110 provided in Figure 2, it will be appreciated that the caching daemon 210 may also couple to multiple directory servers 110 without departing from the spirit and scope of the present invention. Only one directory server 110, however, is depicted in Figure 2 for simplicity sake in illustrating the present invention. The LDAP directory server(s) 110 access data from an LDAP directory (not shown).

Please replace the paragraph beginning at page 9, line 17 through page 10, line 2, with the following rewritten paragraph:

A3
According to one embodiment of the present invention, the caching daemon 210 operates transparently from the perspective of the client server applications 120 that desire to access data from the directory server 110. To integrate the caching daemon 210 into the OS upon which these applications 120 are running, the SIA layer 220 is used to create a shared library with a predefined set of Application Programming Interfaces (APIs). The APIs correspond to the required set of security APIs that are used by the OS. By creating this shared library, it causes all of the existing libc security APIs to resolve to the new library. Accordingly, by using the SIA layer 220, all of the "security aware" applications 120 running on the OS will directly access the caching daemon 210 in a transparent manner, i.e., the client server applications 120 will not realize they are accessing the caching daemon 210, but rather will believe they are accessing the directory server 110 directly [~~Dave, can you provide more details on this~~].